Consultative
Committee for
Space Data Systems

RECOMMENDATION FOR SPACE
DATA SYSTEM STANDARDS

PACKET TELEMETRY

CCSDS 102.0-B-3

BLUE BOOK

NOVEMBER 1992

AUTHORITY

Issue:	Blue Book, Issue 3
Date:	November 1992
Location:	Annapolis, Maryland, USA

This Recommendation reflects the consensus technical agreement of the following Member Agencies of the **CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS** (CCSDS):

- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centre National D’Etudes Spatiales (CNES)/France.
- Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V.(DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.

The following Observer Agencies also concur with this Recommendation:

- Chinese Academy of Space Technology (CAST)/People’s Republic of China.
- Central Research Institute of Physics (CRIP)/Hungary.
- Department of Communications, Communications Research Centre (DOC-CRC)/Canada.
- Institute of Space and Astronautical Science (ISAS)/Japan.

This Recommendation is published and maintained by:

CCSDS Secretariat
Program Integration Division (Code-OI)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

The **CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS (CCSDS)** is an organisation officially established by the management of member space Agencies. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **RECOMMENDATION**s and are not considered binding on any Agency.

This **RECOMMENDATION** is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this **RECOMMENDATION** is entirely voluntary. Endorsement, however, indicates the following understandings:

— Whenever an Agency establishes a CCSDS-related STANDARD, this STANDARD will be in accord with the relevant **RECOMMENDATION**. Establishing such a STANDARD does not preclude other provisions which an Agency may develop.

— Whenever an Agency establishes a CCSDS-related STANDARD, the Agency will provide other CCSDS member Agencies with the following information:

  • The STANDARD itself.

  • The anticipated date of initial operational capability.

  • The anticipated duration of operational service.

— Specific service arrangements shall be made via memoranda of agreement. Neither this **RECOMMENDATION** nor any ensuing STANDARD is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **RECOMMENDATION** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or cancelled.

In those instances when a new version of a **RECOMMENDATION** is issued, existing CCSDS-related Agency standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each Agency to determine when such standards or implementations are to be modified. Each Agency is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the **RECOMMENDATION**.

FOREWORD

This document is a technical **RECOMMENDATION** for use in developing packetised telemetry systems and has been prepared by the **CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS** (CCSDS). The Packet Telemetry concept described herein is the baseline concept for spacecraft-to-ground data communication within missions that are cross-supported between Agencies of the CCSDS.

This **RECOMMENDATION** establishes a common framework and provides a common basis for the data structures of spacecraft telemetry streams. It allows implementing organisations within each Agency to proceed coherently with the development of compatible derived Standards for the flight and ground systems that are within their cognizance. Derived Agency Standards may implement only a subset of the optional features allowed by the **RECOMMENDATION** and may incorporate features not addressed by the **RECOMMENDATION**.

Through the process of normal evolution, it is expected that expansion, deletion or modification to this document may occur. This **RECOMMENDATION** is therefore subject to CCSDS document management and change control procedures which are defined in Reference [1].

DOCUMENT CONTROL

A. FIRST ISSUE

DOCUMENT REFERENCE:	CCSDS 102.0-B-1
TITLE:	Recommendation for Space Data System Standards: Packet Telemetry, Issue 1
DATE:	May 1984

B. ISSUE 2

DOCUMENT REFERENCE:	CCSDS 102.0-B-2
TITLE:	Recommendation for Space Data System Standards: Packet Telemetry, Issue 2
DATE:	January 1987

UPDATES:

Substantive technical changes from the previous issue had been flagged with change bars in the margin

C. ISSUE 3

DOCUMENT REFERENCE:	CCSDS 102.0-B-3
TITLE:	Recommendation for Space Data System Standards: Packet Telemetry, Issue 3
DATE:	November 1992

UPDATES:

A distinct marking of all changes was not practical, because the RECOMMENDATION has been re-written in a different style without changing its technical contents except as noted below. This text provides concise information on the changes. More detailed information can be found in Reference [5].

a. Changes not Compatible with the Previous Issue

Special attention should be paid to the “Statement of Intent” on page ii of this document when reading the changes listed under this category.

Source Packet:

(1) The first bit of the Packet Secondary Header is no longer reserved to distinguish between CCSDS-standard and CCSDS-non-standard Secondary Headers.

Source Packet Segment:

(2) The value “11” for the Segmentation Flags is no longer allowed for Source Packet Segments (called “Telemetry Segments” in the previous issues).

Transfer Frame:

(3) Option B, the option of including the Attached Synchronisation Marker in the data space covered by the Frame Error Control Field, is no longer provided.
(4) The reverse order insertion of Source Packets or Segments has been eliminated.

b. Enhancements and other Changes Compatible with the Previous Issue

Source Packet:

(5) An option for defining Groups of Source Packets has been added (see Item 3.e).
(6) A data structure for the optional Source Packet Secondary Header has been specified (see Sub-Section 3.2.1).
(7) The reference to Source Packet Error Control has been eliminated, as this is application specific.

Source Packet Segment:

(8) The constant length of a Segment is allowed to change on a “Mission Phase” basis (see Item 5.1.5.4.e).

Transfer Frame:

(9) All references to telemetry channel coding—including the items related to synchronisation markers—have been eliminated. Instead the CCSDS Recommendation on Telemetry Channel Coding (Reference [1]) has been referenced.
(10) The concept of a “Master Channel” has been introduced (see Item 5.d).
(11) The constant length of a Transfer Frame is allowed to change on a “Mission Phase” basis (see Item 5.c).
(12) Presence or absence of the Secondary Header, previously fixed for a Physical Channel, is now fixed only for a specific Virtual Channel or for a specific Master Channel (see Paragraph 5.1.5.1).
(13) Presence or absence of the Operational Control Field, previously fixed for a Physical Channel, is now fixed only for a specific Master Channel or specific Virtual Channels throughout a Mission Phase (see Item 5.1.2.3.c).

(14) The allowed contents of the Transfer Frame Data Field has been generalised to include any privately defined data whenever the Synchronisation Flag is set to “1” (see Item 5.a).

(15) A Type-2 Operational Control Field Report can be used (see Section 5.4).

c. Editorial

General:

(16) The document has been written in a new terse style; specifications and comments have been separated; along with these changes a number of editorial modifications have been made.

Source Packet:

(17) The “Segmentation Flags” have been renamed “Grouping Flags”.

(18) The “Packet Length” has been renamed “Packet Data Length”.

(19) The term “Packet Data Field” has been introduced for the data entity consisting of the Secondary Header plus the Source Data Field (see Item 3.b).

Source Packet Segment:

(20) The “Telemetry Segment” has been renamed “Source Packet Segment”.

(21) The “Segment Length”, contained in the Segment Header, has been renamed “Residual Packet Length”.

Transfer Frame:

(22) The “Frame Error Control Word” has been renamed the “Frame Error Control Field”.

CONTENTS

FIGURES

REFERENCES

[1] "Procedures Manual for the Consultative Committee for Space Data Systems", CCSDS A00.0-Y-5, May 1992, or later issue.

[2] "Telemetry Channel Coding", Blue Book, CCSDS 101.0-B-3, May 92, or later issue.

[3] "Time Code Formats", Blue Book, CCSDS 301.0-B-2, April 90, or later issue.

[4] "Telecommand, Part 2: Data Routing Service", Blue Book, CCSDS 202.0-B-2, November 92, or later issue.

[5] "Telemetry: Concept and Rationale", Green Book, CCSDS 100.0-G-1, December 87, or later issue.

[6] "Advanced Orbiting Systems, Networks and Data Links: Architectural Specification", Blue Book, CCSDS 701.0-B-2, November 92, or later issue.

These documents may be obtained from the CCSDS Secretariat at the address indicated on page i.

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a common **RECOMMENDATION** for the implementation of spacecraft "Packet Telemetry" systems by the Agencies participating in the **CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS (CCSDS)**.

1.2 SCOPE

PACKET TELEMETRY is a concept which facilitates the transmission of space-acquired data from source to user in a standardised highly automated manner. **PACKET TELEMETRY** provides a mechanism for implementing common data transport structures and protocols which may enhance the development and operation of space mission systems.

This **RECOMMENDATION** addresses the following two processes:

— The end-to-end transport of space mission data sets from source application processes located in space to distributed user application processes located on the ground.

— The intermediate transfer of these data sets through space data acquisition networks, which contain spacecraft, radio links, tracking stations, ground communications circuits and mission control centres as some of their components.

This **RECOMMENDATION** is limited to describing the telemetry formats which are generated by the spacecraft in order to execute its role in the above processes. The CCSDS channel coding and synchronisation mechanisms required to implement space-to-ground data links of acceptable quality are defined in Reference [2].

An overview of the **PACKET TELEMETRY** Concept is given in Chapter 2.

1.3 APPLICABILITY

This **RECOMMENDATION** applies to the creation of Agency standards and to the future exchange of **PACKET TELEMETRY** between CCSDS Agencies in cross-support situations. The **RECOMMENDATION** includes comprehensive specification of the structure of data streams that are generated by remote space vehicles for telemetering to space mission data processing facilities (which are usually located on Earth). The **RECOMMENDATION** does not attempt to define the architecture or configuration of these data processing facilities, except to describe assumed ground data handling services which affect the selection of certain on-board formatting options.

The **RECOMMENDATION** specifies a wide range of formatting capabilities which may facilitate a high degree of flexibility in the design of spacecraft data acquisition systems; however, compatibility with the **PACKET TELEMETRY** concept may be realised by only implementing a narrow subset of these capabilities. Some "Application Notes" which discuss how different levels of compatibility may be achieved are included in Reference [5].

1.4 RATIONALE

The CCSDS believes it is important to document the rationale underlying the recommendations chosen, so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The concept and rationale for **PACKET TELEMETRY** may also be found in Reference [5].

1.5 STRUCTURE OF THE DOCUMENT

For the designation of text partitions the following conventions will be used:

— text designated by one number belongs to a Chapter;
— text designated by two numbers belongs to a Section;
— text designated by three numbers belongs to a Sub-Section;
— text designated by four numbers belongs to a Paragraph;
— text designated by a lower case letter belongs to an Item.

All specifications are contained in the Chapters 3, 4 and 5 of this **RECOMMENDATION**. They are identified by an Item Number consisting of the number of the text partition as defined above, and a lower case letter. The conventions and definitions applied in these specifications are itemised in Section 1.6.

All other text and all figures in these chapters represent comments to these specifications. All comments are printed in italics.

The contents of the specifications take precedence over those of the comments.

All terms printed in bold-face upper-case are referenced in the Index.

1.6 CONVENTIONS AND DEFINITIONS

The following Items contain the conventions which have been used throughout this **RECOMMENDATION**.

a. To identify each bit in an **N-BIT FIELD** the first bit in the field to be transferred (i.e., the most left justified when drawing a figure) is defined to be "Bit 0"; the following bit is defined to be "Bit 1" and so on up to "Bit N-1". When the field is used to express a binary value (such as a counter), the **MOST SIGNIFICANT BIT** shall be the first bit of the field, i.e., "Bit 0" (see Figure 1-1).

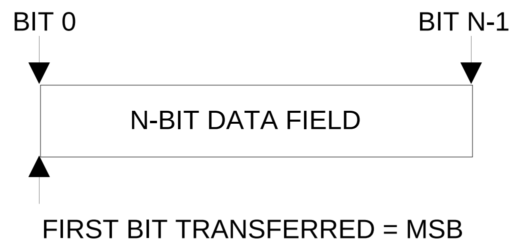


FIGURE 1-1: BIT NUMBERING CONVENTION

b. In accordance with modern data communication practice, spacecraft data fields are often grouped into 8-bit words which conform to convention 1.6.a. Throughout this **RECOMMENDATION**, such an 8-bit word is termed **OCTET**.

c. The numbering for **OCTET**s within a data structure starts with 0.

d. The term **MISSION PHASE** designates a period of a mission during which specified telemetry characteristics are fixed. The transition between two consecutive **MISSION PHASE**s may cause an interruption of the telemetry services.

e. Certain characteristics of the data structures specified in this **RECOMMENDATION** are required to remain unchanged throughout a **MISSION PHASE** or throughout all **MISSION PHASE**s. In these cases the term "static" is used to specify characteristics which remain unchanged either with respect to an **APPLICATION PROCESS IDENTIFIER** (for definition see Paragraph 3.1.2.3), or within a specific **VIRTUAL CHANNEL** (for definition see Item 5.e) or within a specific **MASTER CHANNEL** (for definition see Item 5.d).

f. **IDLE DATA** is data which carries no information, but is sent to meet timing or synchronisation requirements. The bit pattern of **IDLE DATA** is not specified.

2. OVERVIEW

This **PACKET TELEMETRY RECOMMENDATION** describes data structures used to transport data from data sources on board a space vehicle to data sinks on the ground, as shown in Figure 2-1.

FIGURE 2-1: CCSDS PACKET TELEMETRY DATA SYSTEM

2.1 THE PACKET TELEMETRY CONCEPT

The essence of the packet telemetry concept is to permit multiple application processes running in on-board sources to create units of data as best suits each data source, and then to permit the on-board data system to transmit these data units over a space-to-ground communications channel in a way that enables the ground system to recover the individual data units with high reliability and provide them to the data sinks in sequence. These on-board sources are either instruments or sub-systems.

To accomplish these functions, this Recommendation defines two data structures—**SOURCE PACKET**s and **TRANSFER FRAME**s—and a multiplexing process to interleave **SOURCE PACKET**s from various **APPLICATION PROCESS**es into **TRANSFER FRAME**s.

2.2 SOURCE PACKET

The **SOURCE PACKET**, which in the following text may also be termed packet, is a data structure generated by an on-board **APPLICATION PROCESS** in a way that is responsive to the needs of that process. It can be generated at fixed or variable intervals and may be fixed or variable in length. Aside from a packet header that identifies the source and characteristics of the packet, the internal data content of the **SOURCE PACKET** is completely under the control of the **APPLICATION PROCESS**.

The **SOURCE PACKET** allows each **APPLICATION PROCESS** within a data source to optimise the size and structure of its data set with a minimum of constraints imposed by the spacecraft-to-ground transport system. Each data source is thus able to define its data organisation independently of other data sources and to adapt this organisation to the various modes of the instrument or sub-system.

The **SOURCE PACKET PRIMARY HEADER** contains an **APPLICATION PROCESS IDENTIFIER** used to route the packet to its destination sink. The header also carries information about the length, sequence, and other characteristics of the packet. An optional **SOURCE PACKET SECONDARY HEADER** is provided for standardised time-tagging of **SOURCE PACKET**s, and to carry application-unique ancillary data.

An optional mechanism is provided for transmitting long **SOURCE PACKET**s as a series of shorter packet segments, thus avoiding exclusive capture of the channel by one source. The data structure that provides this feature is the **SOURCE PACKET SEGMENT**, which has a structure similar to the **SOURCE PACKET**.

2.3 TRANSFER FRAME

The **TRANSFER FRAME** is a data structure that provides an envelope for transmitting packetised data over a noisy space-to-ground channel. It carries information in the **TRANSFER FRAME PRIMARY HEADER** that permits the ground system to route the **TRANSFER FRAME**s to their intended destination. The **TRANSFER FRAME** is of fixed length (for a given **PHYSICAL CHANNEL** during a **MISSION PHASE**). It is compatible with the CCSDS Recommendation for Telemetry Channel Coding (including synchronisation), (see Reference [2]); thus the transmitted data can be recovered with extremely high reliability.

Multiple, individual, asynchronous **APPLICATION PROCESS**es on board a space vehicle can generate variable-length **SOURCE PACKET**s at different rates, and these **SOURCE PACKET**s can then be multiplexed together into a synchronous stream of fixed-length coded **TRANSFER FRAME**s for reliable transmission to the ground.

The **TRANSFER FRAME PRIMARY HEADER** provides the necessary elements to allow the variable-length **SOURCE PACKET**s and/or **SOURCE PACKET SEGMENT**s from a number of **APPLICATION PROCESS**es on a spacecraft to be multiplexed into a sequence of fixed-length frames. Short packets may be contained in a single frame, while longer ones may span two or more frames. Since a packet can begin or end at any place in a frame, the entire data field of every frame can be used to carry data; there is no need to tune the sizes of packets or their order of occurrence to fit the frames.

A mechanism (**IDLE PACKET**s) is provided for cases where a frame must be released and insufficient packet data is available. Further, frames containing **IDLE DATA** are defined to keep the data capture element in synchronisation in the absence of data.

On the ground, the information in the frame and packet headers allows the data acquisition system to extract packets in a standardised way, with a data-independent method of reassembling **SOURCE PACKET**s from **SOURCE PACKET SEGMENT**s and determining data completeness.

In addition to packets, the **TRANSFER FRAME** can carry two optional fields, the **TRANSFER FRAME SECONDARY HEADER** and the **OPERATIONAL CONTROL FIELD**. The **TRANSFER FRAME SECONDARY HEADER** can be used to carry fixed-length mission-specific data. The **OPERATIONAL CONTROL FIELD** can be used to provide the status of telecommand or other spacecraft operations activities. Instead of packets the **TRANSFER FRAME** can carry **PRIVATELY DEFINED DATA**.

2.4 SHARING TRANSMISSION RESOURCES

As most space communication systems are capacity-limited, multiple users must share access to the downlink data channel, so the on-board data system must be able to manage the data flow to the ground in an orderly manner. In addition, different types of data may be handled differently on the spacecraft or on the ground. This Recommendation provides two methods for controlling data flow, namely:

(1) **VIRTUAL CHANNELISATION**, a mechanism that allows the various sources which generate packets to be "virtually" given exclusive access to this **PHYSICAL CHANNEL** by assigning them transmission capacity on a frame-by-frame basis. Each **TRANSFER FRAME** is identified as belonging to one of the up to eight **VIRTUAL CHANNEL**s. **VIRTUAL CHANNELISATION** is normally used to separate sources or destinations with different characteristics. For example, if a payload contains an imaging instrument which produces packets containing many thousands of octets, and a number of other instruments which generate smaller packets, a possible system architecture would be to assign the imaging instrument packets to one **VIRTUAL CHANNEL** and to handle the rest by multiplexing them onto a second **VIRTUAL CHANNEL**. **VIRTUAL CHANNEL**s may also be used to separate real-time packets from recorded packets, both on the spacecraft and on the ground, and to allow easy separation on the ground of data streams that are to be sent to different destinations.

(2) **SOURCE PACKET SEGMENTATION**, in which very long **SOURCE PACKET**s generated by **APPLICATION PROCESS**es are subdivided by the on-board data system into smaller, fixed-length **SOURCE PACKET SEGMENT**s. **SOURCE PACKET SEGMENT**s from multiple data sources may then be multiplexed with smaller **SOURCE PACKET**s into a single **VIRTUAL CHANNEL**, thus providing shared access to the capacity of that **VIRTUAL CHANNEL**. The original **SOURCE PACKET**s are later reconstructed on the ground, using information in the **SOURCE PACKET SEGMENT** and **TRANSFER FRAME** headers.

Figure 2-2 shows the flow of telemetry data from several on-board sources (instruments or sub-systems), through to the delivery of the same data to **SINK PROCESS**es on the ground. At the top of the figure, generation of **SOURCE PACKET**s from **APPLICATION PROCESS**es in several data sources is shown, as well as segmentation of packets from some of the sources. These packets and segments are multiplexed into the **TRANSFER FRAME**s of several **VIRTUAL CHANNEL**s. These **TRANSFER FRAME**s are transmitted to the ground, using appropriate error protection and synchronisation techniques. On the ground they are demultiplexed into **VIRTUAL CHANNEL**s, and the packets or segments are extracted. Data from **SOURCE PACKET SEGMENT**s are extracted and combined to reconstruct the original **SOURCE PACKET**. **SOURCE PACKET**s are then delivered to **SINK PROCESS**es, shown at the bottom of the Figure, using the **APPLICATION PROCESS IDENTIFIER**s in the **SOURCE PACKET** headers for routing. **SOURCE PACKET**s with a given **APPLICATION PROCESS IDENTIFIER** may be delivered to one or more **SINK PROCESS**es. Packets may be time-ordered prior to delivery using the information in the **PACKET PRIMARY HEADER** and the **PACKET SECONDARY HEADER**.

2.5 APPLICATION NOTES

Application Notes, which describe how compatibility with these various data structures may be achieved, are presented in Reference [5], along with key elements of the rationale behind **PACKET TELEMETRY**.

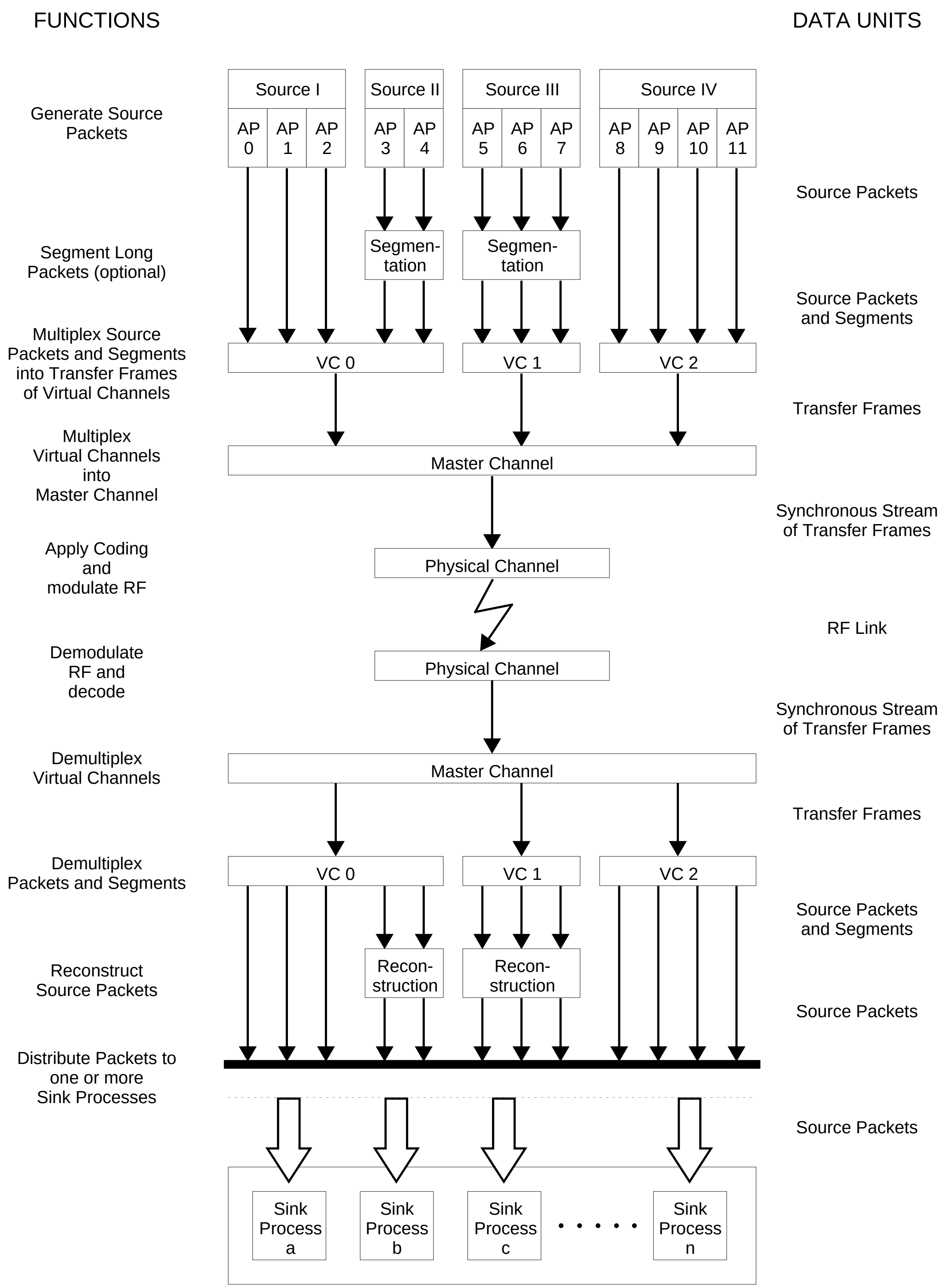


FIGURE 2-2: TELEMETRY DATA FLOW

3. SOURCE PACKET

a. A **SOURCE PACKET**, which in the following text may also be termed **PACKET**, shall encapsulate a block of observational and ancillary application data which is to be transmitted from an **APPLICATION PROCESS** in space to one or several **SINK PROCESS**es on the ground.

b. The **SOURCE PACKET** shall consist of two major fields, positioned contiguously, in the following sequence:

	Length in bits
— **PACKET PRIMARY HEADER** (mandatory)	48
— **PACKET DATA FIELD** (mandatory)	variable

c. The **SOURCE PACKET** shall consist of at least 7 and at most 65542 octets.

d. A **SOURCE PACKET** which contains **IDLE DATA** in its **PACKET DATA FIELD** is called an **IDLE PACKET**.

Idle Packets may be generated by the on-board data system when needed to maintain synchronisation of the data transport and the packet extraction processes.

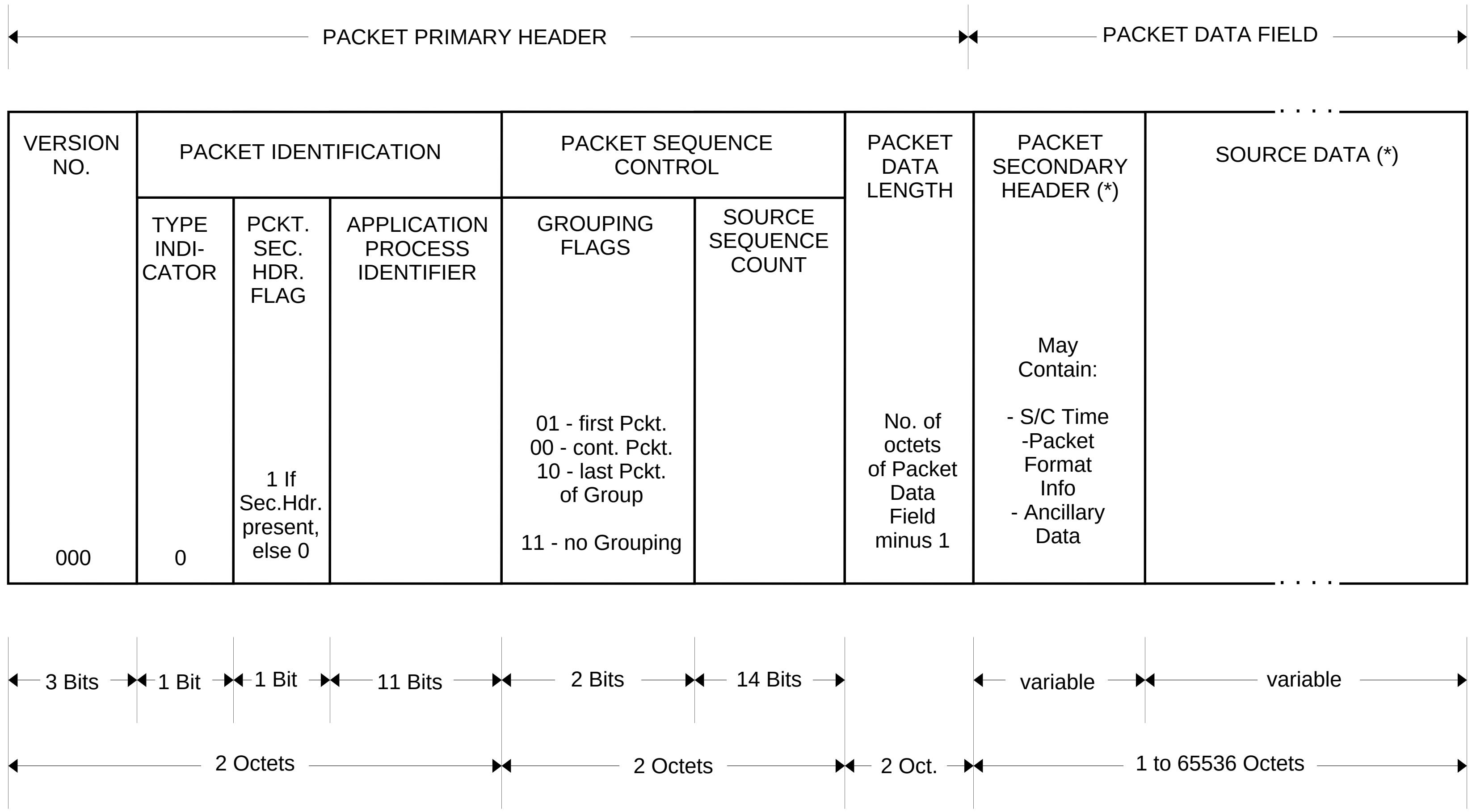
e. A series of **SOURCE PACKET**s generated consecutively by a single **APPLICATION PROCESS** may be designated as a **GROUP OF SOURCE PACKETS**.

Figure 3-1 shows the format of the Source Packet as specified above including the sub-formats to be specified in the following sections.

3.1 PACKET PRIMARY HEADER

a. The **PACKET PRIMARY HEADER** is mandatory and shall consist of the four fields, positioned contiguously, in the following sequence:

	Length in bits
— **VERSION NUMBER**	3
— **PACKET IDENTIFICATION**	13
— **PACKET SEQUENCE CONTROL**	16
— **PACKET DATA LENGTH**	16



(*) may or may not be required; for details see specifications in text.

FIGURE 3-1: SOURCE PACKET FORMAT

3.1.1 VERSION NUMBER

a. The **VERSION NUMBER** shall be contained within the bits 0–2 of the **PACKET PRIMARY HEADER**.

b. This 3-bit field shall identify the data unit as a **SOURCE PACKET** and shall be set to “000”.

The Version Number is used for two purposes:

— *the distinction between Source Packets and Source Packet Segments, the latter being specified in Chapter 4 below;*
— *to reserve the possibility of introducing other data structures.*

3.1.2 PACKET IDENTIFICATION FIELD

a. The **PACKET IDENTIFICATION FIELD** shall be contained within the bits 3–15 of the **PACKET PRIMARY HEADER**.

b. This 13-bit field shall be separated into three sub-fields:

	Length in bits
— **TYPE INDICATOR**	1
— **PACKET SECONDARY HEADER FLAG**	1
— **APPLICATION PROCESS IDENTIFIER**	11

The Packet Identification verifies the type of the packet (Telemetry Source Packet), indicates whether the packet carries a Secondary Header or not, and provides information on the source of the data, i.e., the Application Process.

3.1.2.1 TYPE INDICATOR

a. Bit 3 of the **PACKET PRIMARY HEADER** shall contain the **TYPE INDICATOR** indicating the type of data unit.

b. The **TYPE INDICATOR** shall be set to “0”.

Because CCSDS telecommand uses a similar packet structure, the type indicator distinguishes between telemetry and telecommand data units (for telecommand packets the type indicator will be set to “1”, see Reference [4]).

3.1.2.2 PACKET SECONDARY HEADER FLAG

a. Bit 4 of the **PACKET PRIMARY HEADER** shall contain the **PACKET SECONDARY HEADER FLAG**.

b. The **PACKET SECONDARY HEADER FLAG** shall indicate the presence or absence of the **PACKET SECONDARY HEADER** within this **SOURCE PACKET**. It shall be “1”, if a **PACKET SECONDARY HEADER** is present; it shall be “0”, if a **PACKET SECONDARY HEADER** is not present.

c. The **PACKET SECONDARY HEADER FLAG** shall be static with respect to the **APPLICATION PROCESS IDENTIFIER** throughout a **MISSION PHASE**.

d. The **PACKET SECONDARY HEADER FLAG** shall be set to “0” for **IDLE PACKET**s.

3.1.2.3 APPLICATION PROCESS IDENTIFIER

a. Bits 5–15 of the **PACKET PRIMARY HEADER** shall contain the **APPLICATION PROCESS IDENTIFIER**.

b. The **APPLICATION PROCESS IDENTIFIER** shall be different for different **APPLICATION PROCESS**es on the same **MASTER CHANNEL** (for the definition of the **MASTER CHANNEL** see Item 5.d).

c. For **IDLE PACKET**s the **APPLICATION PROCESS IDENTIFIER** shall be “11111111111”, i.e., “all ones”.

This identifier is tailored to local mission needs and is therefore assigned by mission management. Users should note that ground data accounting considerations may limit the number of different Application Processes which may be active simultaneously. Certain Application Process Identifiers have been reserved in Reference [6] to be used for specific purposes.

3.1.3 PACKET SEQUENCE CONTROL FIELD

a. The **PACKET SEQUENCE CONTROL FIELD** shall be contained within bits 16–31 of the **PACKET PRIMARY HEADER**.

b. This 16-bit field shall be sub-divided into two sub-fields as follows:

	Length in bits
— **GROUPING FLAGS**	2
— **SOURCE SEQUENCE COUNT**	14

The Packet Sequence Control Field provides a sequential count of the packets generated with the same Application Process Identifier, and if the grouping feature is applied, provides information on the position of a Source Packet in a group.

3.1.3.1 GROUPING FLAGS

a. Bits 16 and 17 of the **PACKET PRIMARY HEADER** shall contain the **GROUPING FLAGS**.

b. The **GROUPING FLAGS** shall be set as follows:

— “01” for the first **SOURCE PACKET** of a group;
— “00” for a continuing **SOURCE PACKET** of a group;
— “10” for a last **SOURCE PACKET** of a group.

c. For a **SOURCE PACKET** not belonging to a **GROUP OF SOURCE PACKETS** the **GROUPING FLAGS** shall be set to “11”.

In Source Packet Segmentation, which is described in Chapter 4, the Grouping Flags are used as Segmentation Flags. Therefore, grouping of Source Packets and Segmentation of these Packets are mutually exclusive.

It should be noted that the length of Source Packets on Virtual Channels carrying Segments may not exceed LSEGMENT (see Item 5.1.5.4.c; for the definition of LSEGMENT see Item 4.d).

The use of a Group of Source Packets is outside the scope of this Recommendation.

3.1.3.2 SOURCE SEQUENCE COUNT

a. Bits 18–31 of the **PACKET PRIMARY HEADER** shall contain the **SOURCE SEQUENCE COUNT**.

b. The **SOURCE SEQUENCE COUNT** shall provide the sequential binary count of each **SOURCE PACKET** generated by an **APPLICATION PROCESS** identified by a unique **APPLICATION PROCESS IDENTIFIER**.

c. The **SOURCE SEQUENCE COUNT** shall be continuous, modulo 16384.

d. **IDLE PACKET**s are not required to increment the **SOURCE SEQUENCE COUNT**.

e. A re-setting of the **SOURCE SEQUENCE COUNT** before reaching 16383 shall not take place unless it is unavoidable.

The purpose of the field is to order this packet with other packets generated by the same Application Process, even though their natural order may have been disturbed during transport to the user’s processor on the ground.

The field will normally be used in conjunction with a Time Code (see Paragraph 3.2.1.1; its insertion is, however, not mandatory) to provide unambiguous ordering; it is therefore essential that the resolution of the time code is sufficient for this code to increment at least once between successive recyclings of the Source Sequence Count.

If the Source Sequence Count is re-set due to an unavoidable re-initialisation of a process the completeness of a sequence of Source Packets cannot be determined.

3.1.4 PACKET DATA LENGTH FIELD

a. The **PACKET DATA LENGTH FIELD** shall be contained within bits 32–47 of the **PACKET PRIMARY HEADER**.

b. This 16-bit field shall contain a binary number equal to the number of octets in the **PACKET DATA FIELD** minus 1.

c. The value contained in the **PACKET DATA LENGTH FIELD** may be variable and shall be in the range of 1 to 65536 octets.

d. Further constraints on the length of the **PACKET DATA FIELD** are specified in Paragraph 5.1.5.4.

Users should recognise that although very long packets are permissible, these may present special problems in terms of data link monopolisation, source data buffering and network accountability during transfer across the unique channel from the spacecraft to the ground and may add complexity to ground processing. The Recommendation therefore provides the means to segment large Source Packets (see Chapter 4) or to assign these packets to individual Virtual Channels (see Chapter 5).

3.2 PACKET DATA FIELD

a. The **PACKET DATA FIELD** shall follow, without gap, the **PACKET PRIMARY HEADER**.

b. The **PACKET DATA FIELD** is mandatory and shall consist of at least one of the two fields, positioned contiguously, in the following sequence:

	Length in bits
— **PACKET SECONDARY HEADER**	variable
— **SOURCE DATA FIELD**	variable

c. The **PACKET DATA FIELD** shall contain at least one octet.

3.2.1 PACKET SECONDARY HEADER

a. If present, the **PACKET SECONDARY HEADER** shall follow, without gap, the **PACKET DATA LENGTH FIELD**.

b. The **PACKET SECONDARY HEADER** is mandatory if no **SOURCE DATA FIELD** is present; otherwise it is optional. The presence or absence of a **PACKET SECONDARY HEADER** shall be signalled by the **PACKET SECONDARY HEADER FLAG** within the **PACKET IDENTIFICATION FIELD** (see Paragraph 3.1.2.2).

c. If present, the **PACKET SECONDARY HEADER** shall consist of either

   — a **PACKET SECONDARY HEADER DATA FIELD**;
   — or a **PACKET SECONDARY HEADER TIME CODE FIELD**;
   — or a **PACKET SECONDARY HEADER TIME CODE FIELD** followed by a **PACKET SECONDARY HEADER DATA FIELD**.

   The chosen option shall remain static for a specific **APPLICATION PROCESS IDENTIFIER** throughout all **MISSION PHASE**s.

The purpose of the Secondary Header is to allow (but not require) a CCSDS-defined means for placing ancillary data (time, internal data field format, spacecraft position/attitude, etc.) within a Source Packet.

3.2.1.1 PACKET SECONDARY HEADER TIME CODE FIELD

a. The **PACKET SECONDARY HEADER TIME CODE FIELD** shall consist of an integral number of octets.

b. The **PACKET SECONDARY HEADER TIME CODE FIELD** shall consist of one of the CCSDS segmented binary or unsegmented binary time codes specified in Reference [3].

   The time codes defined in Reference [3] consist of an optional P-Field (Preamble Field), which identifies the time code and its characteristics, and a mandatory T-Field (Time Field). Examples of time codes are CCSDS Unsegmented Time Code and CCSDS Day Segmented Time Code. Examples of characteristics are ambiguity period, epoch, length and resolution.

c. The time code selected shall be static for a given **APPLICATION PROCESS IDENTIFIER** throughout all **MISSION PHASE**s.

d. If the characteristics are allowed to change for an **APPLICATION PROCESS IDENTIFIER**, the **P-FIELD** shall be present.

   If the characteristics are static for an Application Process Identifier, the corresponding P-Field need not be present.

e. The presence or absence of the **P-FIELD** in the **PACKET SECONDARY HEADER TIME CODE FIELD** shall be static for an **APPLICATION PROCESS IDENTIFIER** throughout all **MISSION PHASE**s. If present, it shall immediately precede the **T-FIELD**.

For services such as archiving, sorting, processing and correlation with other data sets, the Source Sequence Count may have to be concatenated with a time field in order to unambiguously identify a packet.

See also the comment concerning time code under Paragraph 3.1.3.2.

3.2.1.2 PACKET SECONDARY HEADER DATA FIELD

a. The **PACKET SECONDARY HEADER DATA FIELD** shall consist of an integral number of octets.

The Data Field may contain any ancillary data necessary for the interpretation of the information contained within the Source Data Field of the Packet. The content and the format of this data are not specified by this Recommendation.

3.2.2 SOURCE DATA FIELD

a. If present, the **SOURCE DATA FIELD** shall follow, without gap, either the **PACKET SECONDARY HEADER** (if a **PACKET SECONDARY HEADER** is present) or the **PACKET DATA LENGTH FIELD** (if a **PACKET SECONDARY HEADER** is not present).

b. The **SOURCE DATA FIELD** is mandatory if no **PACKET SECONDARY HEADER** is present, otherwise it is optional.

c. The **SOURCE DATA FIELD** shall contain either **SOURCE DATA** from an **APPLICATION PROCESS** or **IDLE DATA**.

d. The length of the **SOURCE DATA FIELD** may be variable. It shall contain an integral number of octets. See also the specifications in Sub-Section 3.1.4.

4. SOURCE PACKET SEGMENT

a. The **SOURCE PACKET SEGMENT**, which in the following text may also be termed **SEGMENT**, shall be the data structure used for **SOURCE PACKET SEGMENTATION**.

b. The **SOURCE PACKET SEGMENTATION** process shall be applied to the **PACKET DATA FIELD** of the original **SOURCE PACKET**.

c. The **SEGMENT** shall encapsulate a block of the **PACKET DATA FIELD**. Two or more **SEGMENT**s shall be used to transmit a **SOURCE PACKET**.

d. The **SEGMENT** shall consist of two major fields, positioned contiguously, in the following sequence:

	Length in octets
— **SEGMENT HEADER** (mandatory)	6
— **SEGMENT DATA FIELD** (mandatory)	
• all **SEGMENT**s of a **SOURCE PACKET**, except the last	**LSEGMENT**
• last **SEGMENT** of a **SOURCE PACKET**	**R**, with $1 \leq \mathbf{R} \leq \mathbf{LSEGMENT}$

e. **LSEGMENT** shall be either 256, 512, or 1024 octets. For other items concerning the length of the **SEGMENT** see also Paragraph 5.1.5.4.

f. **SOURCE PACKET SEGMENTATION** shall not be applied to a **SOURCE PACKET** which belongs to a **GROUP OF SOURCE PACKETS** (see also Paragraph 3.1.3.1).

A Source Packet belonging to a Group cannot be segmented because the Grouping Flags would not be transferred through the segmentation and re-assembly process.

g. **SOURCE PACKET SEGMENTATION** shall not be applied to **IDLE PACKET**s.

Therefore, the length of an Idle Packet in a Virtual Channel (for definition see Item 5.e) carrying Segments may not exceed the length specified in Paragraph 5.1.5.4.

Figure 4-1 shows the structure of the Source Packet Segment, including the sub-structures to be specified in the following sections. Figure 4-2 shows an example of the Source Packet Segmentation process.

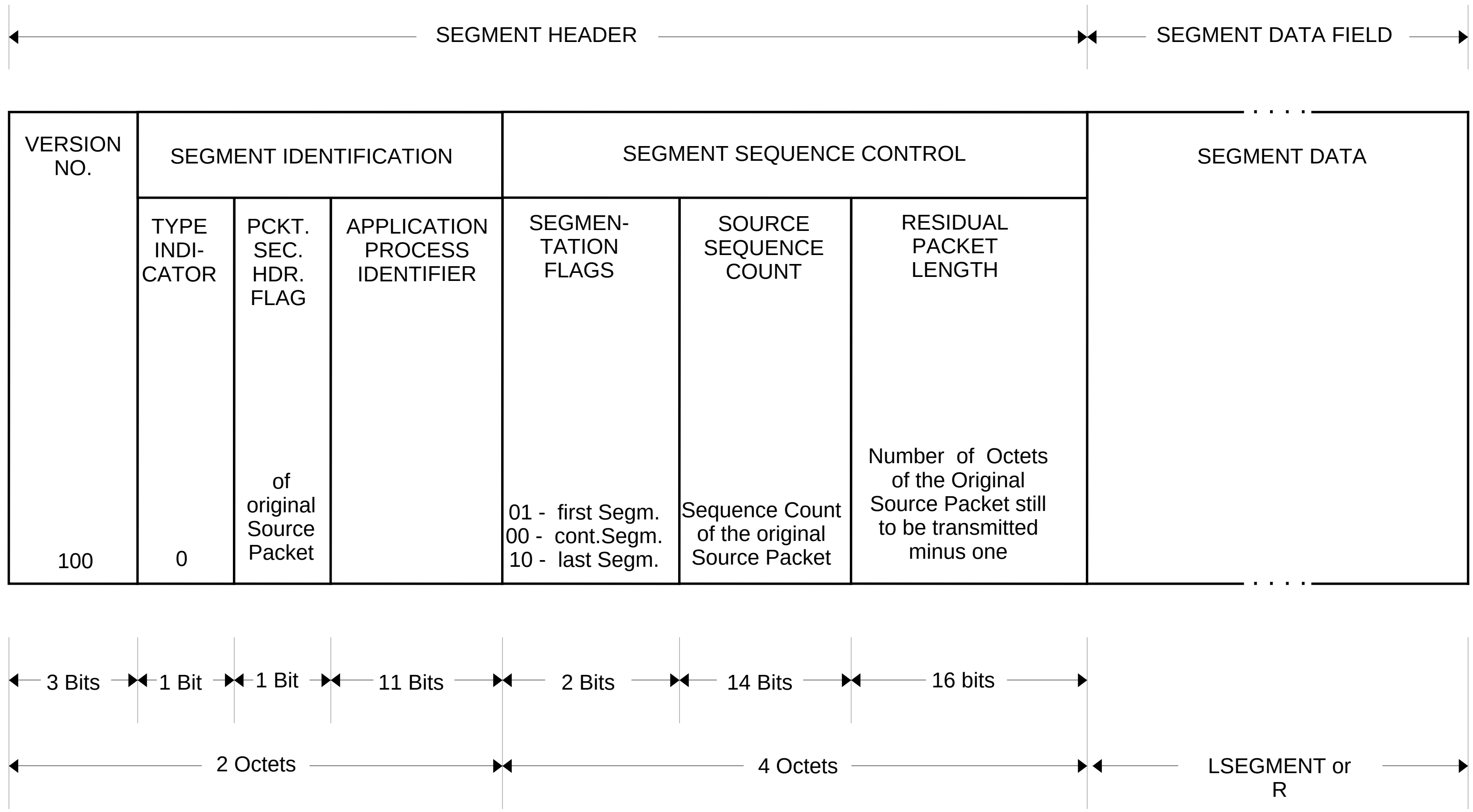


FIGURE 4-1: SOURCE PACKET SEGMENT

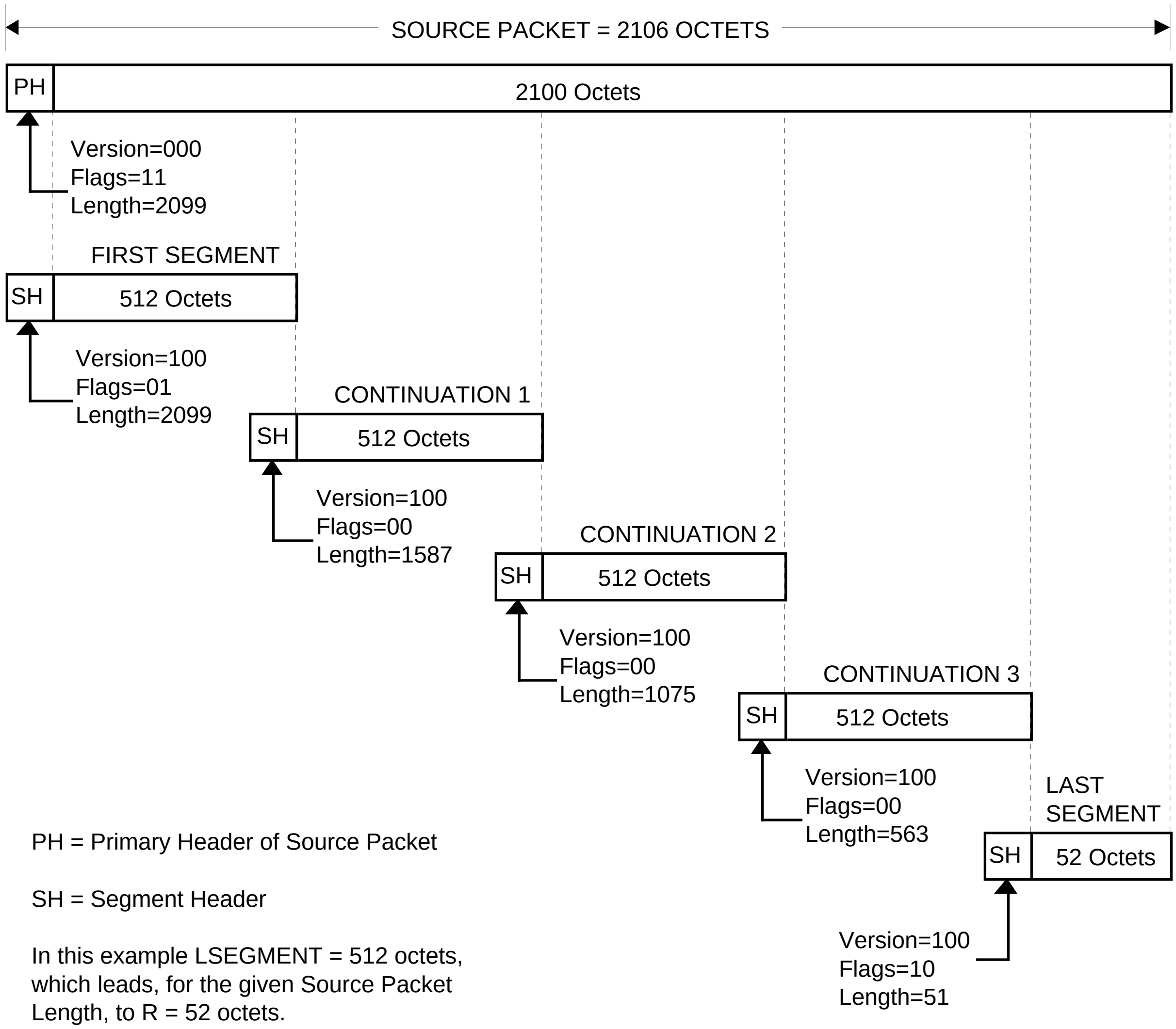


FIGURE 4-2: EXAMPLE OF SOURCE PACKET SEGMENTATION

4.1 SEGMENT HEADER

a. The **SEGMENT HEADER** is mandatory and shall consist of three fields, positioned contiguously, in the following sequence:

	Length in bits
— **VERSION NUMBER**	3
— **SEGMENT IDENTIFICATION**	13
— **SEGMENT SEQUENCE CONTROL**	32

The data structure of the Segment Header and part of its contents are derived from those of the Packet Primary Header.

4.1.1 VERSION NUMBER

a. The **VERSION NUMBER** shall be contained within the bits 0–2 of the **SEGMENT HEADER**.

b. This 3-bit field shall identify the data unit as a **SOURCE PACKET SEGMENT** and shall be set to “100”.

The Version Number is used for two purposes:

— *to distinguish between Source Packet Segments and Source Packets;*
— *to reserve the possibility of introducing other data structures.*

4.1.2 SEGMENT IDENTIFICATION FIELD

a. The **SEGMENT IDENTIFICATION FIELD** shall be contained within bits 3–15 of the **SEGMENT HEADER**.

b. This 13-bit field shall be identical to the **PACKET IDENTIFICATION FIELD** of the original **SOURCE PACKET** in both format and content.

4.1.3 SEGMENT SEQUENCE CONTROL FIELD

a. The **SEGMENT SEQUENCE CONTROL FIELD** shall be contained within bits 16–47 of the **SEGMENT HEADER**.

b. This 32-bit field shall be sub-divided into three sub-fields as follows:

	Length in bits
— **SEGMENTATION FLAGS**	2
— **SOURCE SEQUENCE COUNT**	14
— **RESIDUAL PACKET LENGTH**	16

The Segment Sequence Control Field provides the information necessary to reconstruct the original Source Packet from its constituent Segments.

4.1.3.1 SEGMENTATION FLAGS

a. Bits 16 and 17 of the **SEGMENT HEADER** shall contain the **SEGMENTATION FLAGS**.

b. The **SEGMENTATION FLAGS** shall be set as follows:

— “01” for a **SEGMENT** containing the first block of a **PACKET DATA FIELD**;
— “00” for a **SEGMENT** containing a continuing block of a **PACKET DATA FIELD**;
— “10” for a **SEGMENT** containing the last block of a **PACKET DATA FIELD**.

4.1.3.2 SOURCE SEQUENCE COUNT

a. Bits 18–31 of the **SEGMENT HEADER** shall contain the **SOURCE SEQUENCE COUNT** of the original **SOURCE PACKET**.

4.1.3.3 RESIDUAL PACKET LENGTH

a. The **RESIDUAL PACKET LENGTH FIELD** shall be contained within bits 32–47 of the **SEGMENT HEADER**.

b. The **RESIDUAL PACKET LENGTH FIELD** shall contain the number of octets of the original **SOURCE PACKET DATA FIELD** still to be transmitted, minus one, represented as a binary number. The value in the **RESIDUAL PACKET LENGTH FIELD** includes the octets contained in this **SEGMENT**.

4.2 SEGMENT DATA FIELD

a. The **SEGMENT DATA FIELD** shall follow, without gap, the **RESIDUAL PACKET LENGTH FIELD**.

b. The **SEGMENT DATA FIELD** shall contain one block of the segmented **PACKET DATA FIELD** of the original **SOURCE PACKET**.

c. The **PACKET DATA FIELD** shall be partitioned into **SEGMENT DATA FIELD**s without omissions, fill or overlaps.

d. The length of the **SEGMENT DATA FIELD** shall be **LSEGMENT** except that the length of the last **SEGMENT DATA FIELD** shall be equal to the length **R** of the last block of the original **PACKET DATA FIELD**.

   LSEGMENT is 256, 512 or 1024 octets depending on the value of the Segment Length Identifier in the Transfer Frame Primary Header (see Paragraph 5.1.5.4) for the Virtual Channel (see Paragraph 5.1.2.2) carrying the Segments.

e. The supplementary rules related to **LSEGMENT** provided in Paragraph 5.1.5.4 shall be taken into account.

5. TRANSFER FRAME

a. The **TRANSFER FRAME** shall provide the data structure for the transmission of

(1) **SOURCE PACKET**s,
(2) **SEGMENT**s,
(3) **IDLE DATA** and
(4) **PRIVATELY DEFINED DATA**

across the downlink channel which connects the spacecraft to a data capture element on the ground.

Privately Defined Data may be specialised high rate data or other data not suitable for CCSDS Source Packet structuring.

b. The **TRANSFER FRAME** shall encompass the major fields, positioned contiguously, in the following sequence:

	Length in bits
— **TRANSFER FRAME PRIMARY HEADER** (mandatory)	48
— **TRANSFER FRAME SECONDARY HEADER** (optional)	16, 24, ... or 512
— **TRANSFER FRAME DATA FIELD** (mandatory)	variable
— **OPERATIONAL CONTROL FIELD** (optional)	32
— **FRAME ERROR CONTROL FIELD** (mandatory, if Reed-Solomon Encoding is not applied, otherwise optional)	16

c. The **TRANSFER FRAME** shall be of constant length throughout a specific **MISSION PHASE**. Its length shall be consistent with the specifications contained in Reference [2].

The Telemetry Channel Coding Blue Book (Reference [2]) issued in 1992 limits the Transfer Frame length to 8920 bits.

A change of Frame Length may result in a loss of synchronisation at the data capture element.

d. All **TRANSFER FRAME**s with the same **TRANSFER FRAME VERSION NUMBER** (see Sub-Section 5.1.1) and the same **SPACECRAFT IDENTIFIER** (see Paragraph 5.1.2.1) on the same **PHYSICAL CHANNEL** constitute a **MASTER CHANNEL**.

In most cases the Master Channel will be identical with the Physical Channel. However, in case the Physical Channel also carries Transfer Frames with other Spacecraft Identifiers a distinction between Master Channel and Physical Channel is necessary, i.e., multiplexing of Transfer Frames with different Spacecraft Identifiers will be performed by the multiplexing of different Master Channels on the same Physical Channel.

e. A **MASTER CHANNEL** shall consist of between one and eight **VIRTUAL CHANNEL**s.

Although Packet Telemetry Systems may be designed to tolerate channel noise, full benefit from Packet Telemetry will require that a high-quality data channel is provided so that packetised data may be adaptively inserted into the frame. The relevant CCSDS Recommendation, Reference [2], describes the coding mechanisms for such a channel, including frame synchronisation and randomisation.

Figure 5-1 illustrates the detailed format of the Transfer Frame.

5.1 TRANSFER FRAME PRIMARY HEADER

a. The **TRANSFER FRAME PRIMARY HEADER** is mandatory and shall consist of five fields, positioned contiguously, in the following sequence:

	Length in bits
— **TRANSFER FRAME VERSION NUMBER**	2
— **TRANSFER FRAME IDENTIFICATION**	14
— **MASTER CHANNEL FRAME COUNT**	8
— **VIRTUAL CHANNEL FRAME COUNT**	8
— **TRANSFER FRAME DATA FIELD STATUS**	16

The Primary Header covers five principal functions:

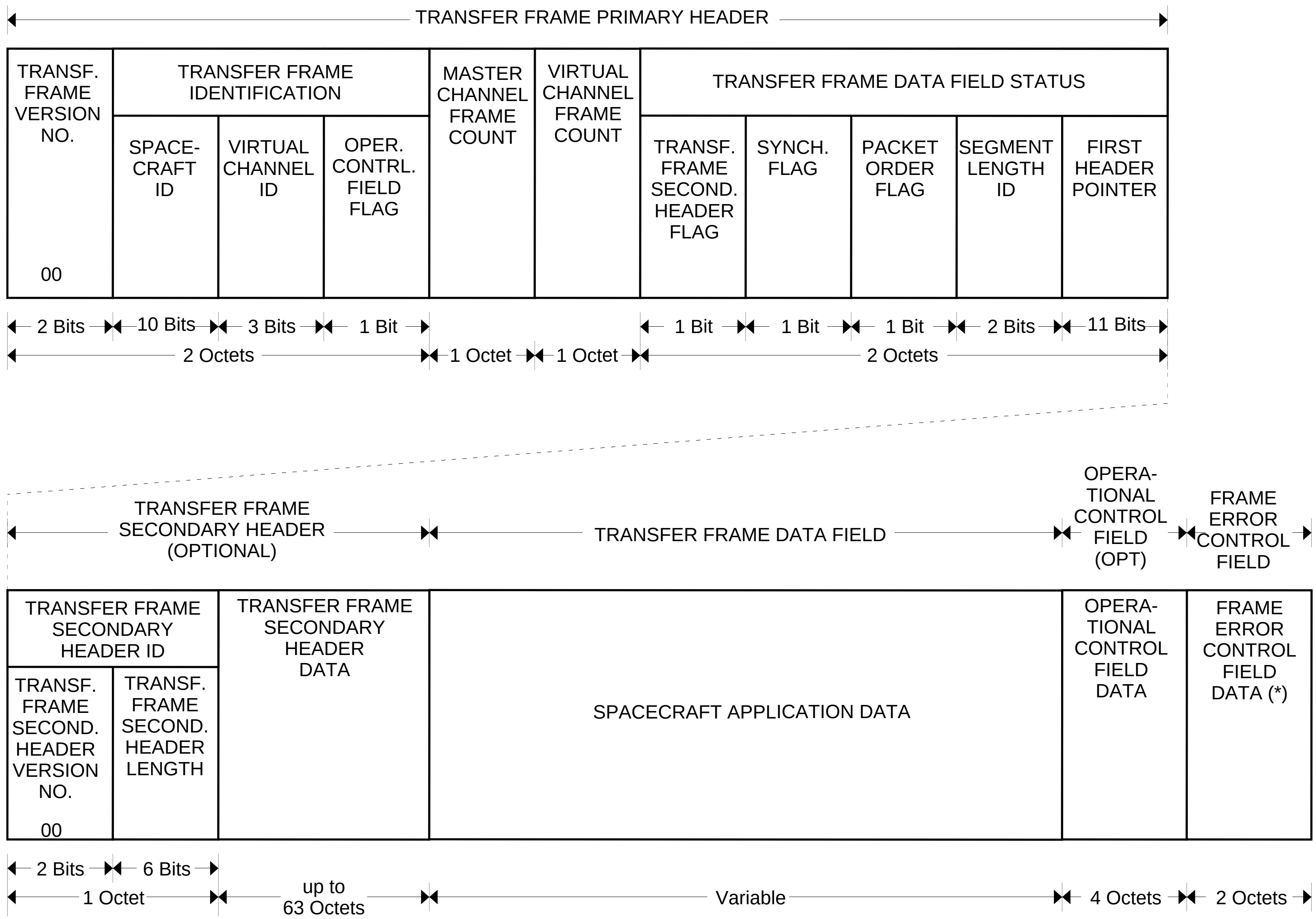
— *Identification of the data unit as a Transfer Frame;*
— *Identification of the spacecraft (and possibly of the link, if applicable), which transmitted the telemetered data;*
— *Multiplexing of the Virtual Channels into one Master Channel;*
— *Providing a counting mechanism for the Virtual Channels and the Master Channel;*
— *Providing pointers and other control information so that variable length Source Packet or Segment data units may be extracted from the Transfer Frame Data Field.*

5.1.1 TRANSFER FRAME VERSION NUMBER

a. The **TRANSFER FRAME VERSION NUMBER** shall be contained within bits 0–1 of the **TRANSFER FRAME PRIMARY HEADER**.

b. This 2-bit field shall identify the data unit as a **TRANSFER FRAME**; it shall be set to “00”.

This Recommendation defines Version 1 of the Transfer Frame. Reference [6] defines a similar data unit which is distinguished by a different value of the Version Number.



(*) may or may not be required; for details see specifications in text.

FIGURE 5-1: TRANSFER FRAME FORMAT

5.1.2 TRANSFER FRAME IDENTIFICATION FIELD

a. The **TRANSFER FRAME IDENTIFICATION FIELD** shall be contained within bits 2–15 of the **TRANSFER FRAME PRIMARY HEADER**.

b. This 14-bit field shall be sub-divided into three sub-fields as follows:

	Length in bits
— **SPACECRAFT IDENTIFIER**	10
— **VIRTUAL CHANNEL IDENTIFIER**	3
— **OPERATIONAL CONTROL FIELD FLAG**	1

This field identifies the generator of the Transfer Frame, it specifies the Virtual Channel to which it belongs, and it provides information on the format of the Transfer Frame.

5.1.2.1 SPACECRAFT IDENTIFIER

a. Bits 2–11 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **SPACECRAFT IDENTIFIER**.

b. The **SPACECRAFT IDENTIFIER** is assigned by CCSDS and shall provide the identification of the spacecraft which created the frame of data.

c. The **SPACECRAFT IDENTIFIER** shall be static throughout all **MISSION PHASE**s.

Different Spacecraft IDs may be assigned for normal operations and for development vehicles using the ground networks during pre-launch test operations, and for simulated data streams. The Secretariat of the CCSDS assigns Spacecraft Identifiers.

5.1.2.2 VIRTUAL CHANNEL IDENTIFIER

a. Bits 12–14 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **VIRTUAL CHANNEL IDENTIFIER**.

b. The **VIRTUAL CHANNEL IDENTIFIER** provides the identification of the **VIRTUAL CHANNEL**.

The order of occurrence of different Virtual Channels on a Master Channel may vary.

5.1.2.3 OPERATIONAL CONTROL FIELD FLAG

a. Bit 15 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **OPERATIONAL CONTROL FIELD FLAG**.

b. The **OPERATIONAL CONTROL FIELD FLAG** shall indicate the presence or absence of the **OPERATIONAL CONTROL FIELD**. It shall be “1”, if the **OPERATIONAL CONTROL FIELD** is present; it shall be “0”, if the **OPERATIONAL CONTROL FIELD** is not present.

c. The **OPERATIONAL CONTROL FIELD FLAG** shall be static either within a specific **MASTER CHANNEL** or specific **VIRTUAL CHANNEL**s throughout a **MISSION PHASE**.

5.1.3 MASTER CHANNEL FRAME COUNT FIELD

a. The **MASTER CHANNEL FRAME COUNT FIELD** shall be contained within bits 16–23 of the **TRANSFER FRAME PRIMARY HEADER**.

b. This 8-bit field shall contain a sequential binary count (modulo 256) of each **TRANSFER FRAME** transmitted within a specific **MASTER CHANNEL**.

c. A re-setting of the **MASTER CHANNEL FRAME COUNT** before reaching 255 shall not take place unless it is unavoidable.

The purpose of this field is to provide a running count of the frames which have been transmitted through the same Master Channel.

If the Master Channel Frame Count is re-set due to an unavoidable re-initialisation, the completeness of a sequence of Transfer Frames cannot be determined.

5.1.4 VIRTUAL CHANNEL FRAME COUNT FIELD

a. The **VIRTUAL CHANNEL FRAME COUNT FIELD** shall be contained within bits 24–31 of the **TRANSFER FRAME PRIMARY HEADER**.

b. This 8-bit field shall contain a sequential binary count (modulo 256) of each **TRANSFER FRAME** transmitted through a specific **VIRTUAL CHANNEL** of a **MASTER CHANNEL**.

c. A re-setting of the **VIRTUAL CHANNEL FRAME COUNT** before reaching 255 shall not take place unless it is unavoidable.

The purpose of this field is to provide individual accountability for each of the maximum eight Virtual Channels, primarily to enable systematic Source Packet and Segment extraction from the Transfer Frame Data Field.

If the Virtual Channel Frame Count is re-set due to an unavoidable re-initialisation the completeness of a sequence of Transfer Frames in the related Virtual Channel can not be determined.

5.1.5 TRANSFER FRAME DATA FIELD STATUS FIELD

a. The **TRANSFER FRAME DATA FIELD STATUS FIELD** shall be contained within bits 32–47 of the **TRANSFER FRAME PRIMARY HEADER**.

b. This 16-bit field shall be sub-divided into five sub-fields as follows:

	Length in bits
— **TRANSFER FRAME SECONDARY HEADER FLAG**	1
— **SYNCHRONISATION FLAG**	1
— **PACKET ORDER FLAG**	1
— **SEGMENT LENGTH IDENTIFIER**	2
— **FIRST HEADER POINTER**	11

This field indicates whether a Secondary Header is present. Further, it provides information on the type of data contained in the frame and provides, together with the Virtual Channel Frame Count, the control information necessary to enable Source Packets or Segments (if present) to be extracted from the Transfer Frame Data Field.

5.1.5.1 TRANSFER FRAME SECONDARY HEADER FLAG

a. Bit 32 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **TRANSFER FRAME SECONDARY HEADER FLAG**.

b. The **TRANSFER FRAME SECONDARY HEADER FLAG** shall signal the presence or absence of the **TRANSFER FRAME SECONDARY HEADER**. It shall be “1”, if a **TRANSFER FRAME SECONDARY HEADER** is present; it shall be “0”, if a **TRANSFER FRAME SECONDARY HEADER** is not present.

c. The **TRANSFER FRAME SECONDARY HEADER FLAG** shall be static within a specific **MASTER CHANNEL** throughout a **MISSION PHASE** when the **TRANSFER FRAME SECONDARY HEADER** is associated with a **MASTER CHANNEL**.

d. The **TRANSFER FRAME SECONDARY HEADER FLAG** shall be static within a specific **VIRTUAL CHANNEL** throughout a **MISSION PHASE** when the **TRANSFER FRAME SECONDARY HEADER** is associated with a **VIRTUAL CHANNEL**.

For the significance of the above mentioned associations see Item 5.2.d.

5.1.5.2 SYNCHRONISATION FLAG

a. Bit 33 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **SYNCHRONISATION FLAG**.

b. The **SYNCHRONISATION FLAG** shall signal the type of data which is inserted into the **TRANSFER FRAME DATA FIELD**. It shall be “0”, if octet-synchronised and forward-ordered **SOURCE PACKET**s/**SEGMENT**s or **IDLE DATA** are inserted; it shall be “1”, if **PRIVATELY DEFINED DATA** is inserted.

c. The **SYNCHRONISATION FLAG** shall be static within a specific **VIRTUAL CHANNEL** throughout a **MISSION PHASE**.

Source Packet/Segment data units are normally inserted into the Transfer Frame Data Field synchronously on octet boundaries one following directly after another. Generally, the Source Packets/Segments “spill over” into the next frame for the same Virtual Channel; therefore, Source Packets/Segments do not usually begin at the first octet of the Transfer Frame Data Field. The location of the first Source Packet/Segment header in a particular Transfer Frame is identified by the First Header Pointer Field (see also comment under Paragraph 5.1.5.5).

If a Project chooses not to observe octet boundaries when placing Source Packet/Segment data into the Transfer Frame Data Field, this is considered to be Privately Defined Data.

5.1.5.3 PACKET ORDER FLAG

a. Bit 34 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **PACKET ORDER FLAG**.

b. If the **SYNCHRONISATION FLAG** is set to “0”, the **PACKET ORDER FLAG** is reserved for future use by the CCSDS and shall be set to “0”.

c. If the **SYNCHRONISATION FLAG** is set to “1”, the use of the **PACKET ORDER FLAG** is undefined.

5.1.5.4 SEGMENT LENGTH IDENTIFIER

a. Bits 35 and 36 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **SEGMENT LENGTH IDENTIFIER**.

b. If a **VIRTUAL CHANNEL** does not support **SOURCE PACKET SEGMENTATION**, the **SEGMENT LENGTH IDENTIFIER** shall be set to “11”.

c. If a **VIRTUAL CHANNEL** supports **SOURCE PACKET SEGMENTATION**, the value of the **SEGMENT LENGTH IDENTIFIER** shall denote the value of **LSEGMENT**; it shall be set as follows:

— “00” for **LSEGMENT** = 256 octets;
— “01” for **LSEGMENT** = 512 octets;
— “10” for **LSEGMENT** = 1024 octets.

The Segment Data Field of the last Segment of a Source Packet may be shorter than LSEGMENT.

d. If a **VIRTUAL CHANNEL** supports **SOURCE PACKET SEGMENTATION** the **PACKET DATA FIELD** of the **SOURCE PACKET**s carried by this **VIRTUAL CHANNEL** shall never be longer than **LSEGMENT**.

e. The **SEGMENT LENGTH IDENTIFIER** shall be static within a specific **VIRTUAL CHANNEL** throughout a **MISSION PHASE**.

f. If the **SYNCHRONISATION FLAG** is set to “1”, the **SEGMENT LENGTH IDENTIFIER** is undefined.

5.1.5.5 FIRST HEADER POINTER

a. Bits 37–47 of the **TRANSFER FRAME PRIMARY HEADER** shall contain the **FIRST HEADER POINTER**.

b. If the **SYNCHRONISATION FLAG** is set to “0”, the **FIRST HEADER POINTER** shall contain information on the position of the first **SOURCE PACKET** or first **SEGMENT** within the **TRANSFER FRAME DATA FIELD**.

c. The locations of the octets in the **TRANSFER FRAME DATA FIELD** shall be numbered in ascending order. The first octet in this field is assigned the number 0. The **FIRST HEADER POINTER** shall contain the binary representation of the location of the first octet of the first **PACKET PRIMARY HEADER** or **SEGMENT HEADER**.

The locations of any subsequent headers within the same Transfer Frame Data Field will be determined by calculating these locations using either the Packet Data Length Field (see Sub-Section 3.1.4) or the Segment Length Identifier (see Paragraph 5.1.5.4) or, for the last Segment, the Residual Packet Length Field.

The specification covers also the following two special cases:

(1) If a first Source Packet or Segment Primary Header starts at the end of the Transfer Frame Data Field within Frame N and spills over into Frame M of the same Virtual Channel, the First Header Pointer in Frame N indicates the start of this Header.

(2) If a Source Packet or Segment Header is split between the Frames N and M (M>N), the First Header Pointer in Frame M ignores the residue of the split header and only indicates the start of any subsequent new Source Packet or Segment Header within Frame M.

In both cases (1) and (2) above, one or more Frames with Idle Data may occur between Frame N and Frame M (M>N).

d. If no **PACKET PRIMARY HEADER** or **SEGMENT HEADER** starts in the **TRANSFER FRAME DATA FIELD**, the **FIRST HEADER POINTER** shall be set to “11111111111”.

e. If a **TRANSFER FRAME** contains **IDLE DATA** in its **TRANSFER FRAME DATA FIELD**, the **FIRST HEADER POINTER** shall be set to “11111111110”.

f. If the **SYNCHRONISATION FLAG** is set to “1”, the **FIRST HEADER POINTER** is undefined.

5.2 TRANSFER FRAME SECONDARY HEADER

a. If present, the **TRANSFER FRAME SECONDARY HEADER** shall follow, without gap, the **TRANSFER FRAME PRIMARY HEADER**.

b. The **TRANSFER FRAME SECONDARY HEADER** is optional; its presence or absence shall be signalled by the **TRANSFER FRAME SECONDARY HEADER FLAG** in the **TRANSFER FRAME PRIMARY HEADER** (see Paragraph 5.1.5.1).

c. The **TRANSFER FRAME SECONDARY HEADER** shall consist of an integral number of octets as follows:

	Length in bits
— **TRANSFER FRAME SECONDARY HEADER IDENTIFICATION FIELD**	8
— **TRANSFER FRAME SECONDARY HEADER DATA FIELD**	8, 16, ... or 504

d. The **TRANSFER FRAME SECONDARY HEADER** shall be associated with either a **MASTER CHANNEL** or a **VIRTUAL CHANNEL**.

The association of a Secondary Header with a Master Channel allows data to be transferred frame-synchronously with respect to this Master Channel.

e. The **TRANSFER FRAME SECONDARY HEADER** shall be of fixed length within the associated **MASTER CHANNEL** or within the associated **VIRTUAL CHANNEL** throughout a **MISSION PHASE**.

5.2.1 TRANSFER FRAME SECONDARY HEADER IDENTIFICATION FIELD

a. The **TRANSFER FRAME SECONDARY HEADER IDENTIFICATION FIELD** shall be contained within bits 0–7 of the **TRANSFER FRAME SECONDARY HEADER**.

b. The **TRANSFER FRAME SECONDARY HEADER IDENTIFICATION FIELD** shall be sub-divided into two sub-fields as follows:

	Length in bits
— the **TRANSFER FRAME SECONDARY HEADER VERSION NUMBER FIELD**	2
— the **TRANSFER FRAME SECONDARY HEADER LENGTH FIELD**	6

5.2.1.1 TRANSFER FRAME SECONDARY HEADER VERSION NUMBER

a. The **TRANSFER FRAME SECONDARY HEADER VERSION NUMBER** shall be contained within bits 0–1 of the **TRANSFER FRAME SECONDARY HEADER** .

b. The **TRANSFER FRAME SECONDARY HEADER VERSION NUMBER** shall be set to “00”.

This sub-field shall indicate which of up to four Secondary Header Versions is used. The present Recommendation recognises only one version.

5.2.1.2 TRANSFER FRAME SECONDARY HEADER LENGTH

a. The **TRANSFER FRAME SECONDARY HEADER LENGTH** shall be contained within bits 2–7 of the **TRANSFER FRAME SECONDARY HEADER** .

b. This sub-field shall contain the total length of the **TRANSFER FRAME SECONDARY HEADER** in octets minus one, represented as a binary number.

c. The **TRANSFER FRAME SECONDARY HEADER LENGTH** shall be static either within a specific **MASTER CHANNEL** or a specific **VIRTUAL CHANNEL** throughout a **MISSION PHASE**.

When a Secondary Header is present, this length may be used to compute the location of the start of the Frame Data Field.

5.2.2 TRANSFER FRAME SECONDARY HEADER DATA FIELD

a. The **TRANSFER FRAME SECONDARY HEADER DATA FIELD** shall follow, without gap, the **TRANSFER FRAME SECONDARY HEADER IDENTIFICATION FIELD**.

b. The **TRANSFER FRAME SECONDARY HEADER DATA FIELD** shall contain the **TRANSFER FRAME SECONDARY HEADER** data.

5.3 TRANSFER FRAME DATA FIELD

a. The **TRANSFER FRAME DATA FIELD** shall follow, without gap, the **TRANSFER FRAME PRIMARY HEADER** or the **TRANSFER FRAME SECONDARY HEADER** if present.

b. The **TRANSFER FRAME DATA FIELD** shall contain the data to be transmitted across the downlink channel and shall consist of an integral number of octets. **TRANSFER FRAME DATA** may be any of the four types of data specified in Item 5.a.

c. **SOURCE PACKET**s and **SEGMENT**s shall be inserted contiguously and in forward order into the **TRANSFER FRAME DATA FIELD**.

   If Source Packets and Segments are not inserted this way, they are treated as Privately Defined Data and have to be flagged accordingly.

d. The length of the **TRANSFER FRAME DATA FIELD** shall be constrained by the length of the total **TRANSFER FRAME**. For this constraint see Item 5.c.

e. **SEGMENTS** and **SOURCE PACKETS** may either be transmitted on separate **VIRTUAL CHANNEL**s or may be mixed on the same **VIRTUAL CHANNEL**.

f. **SOURCE PACKET**s and **SEGMENT**s shall not be mixed with **PRIVATELY DEFINED DATA** on the same **VIRTUAL CHANNEL**.

g. In the case where not sufficient **SOURCE PACKET**s (including **IDLE PACKET**s) or **SEGMENT**s or **PRIVATELY DEFINED DATA** are available to fill a **TRANSFER FRAME DATA FIELD**, a **TRANSFER FRAME** with a data field containing only **IDLE DATA** shall be transmitted.

   Transfer Frames containing Idle Data in their data fields have to be sent to maintain synchronisation with the ground station and also because the Secondary Header and the Operational Control Field may still be needed to transmit valid data.

Transfer Frames carrying Idle Data may be sent on a Virtual Channel that also carries Packets and/or Segments, but it is preferred that a separate Virtual Channel is dedicated to Idle Data.

Idle Data in a Transfer Frame Data Field must not be confused with Idle Packets specified in Item 3.d.

Packets and Segments with different Application Process Identifiers may be multiplexed in the Frame Data Field in any combination. In particular, a Packet or Segment with one Application Process Identifier may be placed between two Segments belonging to a Source Packet with a different Application Process Identifier.

5.4 OPERATIONAL CONTROL FIELD

a. If present, the **OPERATIONAL CONTROL FIELD** shall occupy the four octets following, without gap, the **TRANSFER FRAME DATA FIELD**.

b. The **OPERATIONAL CONTROL FIELD** is optional; its presence or absence is signalled by the **OPERATIONAL CONTROL FIELD FLAG** in the **TRANSFER FRAME PRIMARY HEADER** (see Paragraph 5.1.2.3).

c. If present, the field shall occur within every **TRANSFER FRAME** transmitted either through a specific **MASTER CHANNEL** or specific **VIRTUAL CHANNEL**s throughout a **MISSION PHASE**.

d. The leading bit of the field, i.e., bit 0, shall contain a **TYPE FLAG** with the following meanings:

— the **TYPE FLAG** shall be “0”, if the **OPERATIONAL CONTROL FIELD** holds a **TYPE-1-REPORT** which shall contain a **COMMAND LINK CONTROL WORD** the content of which is defined in Reference [4];
— the **TYPE FLAG** shall be “1”, if the **OPERATIONAL CONTROL FIELD** holds a **TYPE-2-REPORT**.

The Type Flag may vary between Transfer Frames on the same Virtual Channel.

e. The first bit of a **TYPE-2-REPORT** shall indicate the use of this report as follows:

— if this bit is “0”, the contents of the report are project-specific;
— if this bit is “1”, the contents of the report are reserved by CCSDS for future application.

The value of the first bit of a Type-2-Report may vary between Transfer Frames on the same Virtual Channel.

The purpose of this field is to provide a standardised mechanism for reporting a small number of real-time functions (such as telecommand verification or spacecraft clock calibration); currently the use for telecommand verification has been defined by CCSDS (Type-1-Reports). This issue of the Recommendation does not define the use of Type-2-Reports; however, it preserves the possibility to do so in future issues by restricting the utilisation of the first bit.

If Operational Control Fields from the same source are carried in more than one Virtual Channel, it is the responsibility of both the on-board and ground processing to maintain the order of these Operational Control Fields.

5.5 FRAME ERROR CONTROL FIELD

a. If present, the **FRAME ERROR CONTROL FIELD** shall occupy the two octets following, without gap, the **OPERATIONAL CONTROL FIELD** if this is present, or the **TRANSFER FRAME DATA FIELD** if an **OPERATIONAL CONTROL FIELD** is not present.

b. The **FRAME ERROR CONTROL FIELD** is optional if the **TRANSFER FRAME** is synchronously contained within the data space of a **REED-SOLOMON CODE BLOCK**.

c. The **FRAME ERROR CONTROL FIELD** is mandatory if the **TRANSFER FRAME** is not Reed-Solomon encoded.

d. If present, the **FRAME ERROR CONTROL FIELD** shall occur within every **TRANSFER FRAME** transmitted within the same **MASTER CHANNEL** throughout a **MISSION PHASE**.

The purpose of this field is to provide a capability for detecting errors which may have been introduced into the frame during the transmission and data handling process.

5.5.1 ENCODING PROCEDURE

a. The **ENCODING PROCEDURE** accepts an (n-16)-bit **TRANSFER FRAME**, excluding the **FRAME ERROR CONTROL FIELD**, and generates a systematic binary (n,n-16) block code by appending a 16-bit **FRAME ERROR CONTROL FIELD** as the final 16 bits of the codeblock.

b. The equation for the contents of the **FRAME ERROR CONTROL FIELD** is:

$$FECF = [(X^{16} \cdot M(X)) + (X^{(n-16)} \cdot L(X))] \text{ modulo } G(X)$$

where

— all arithmetic is modulo 2;

— n is the number of bits in the encoded message;

— M(X) is the (n-16)-bit message to be encoded expressed as a polynomial with binary coefficients;

— L(X) is the presetting polynomial given by

$$L(X) = \sum_{i=0}^{15} X^{i} \, ;$$

— G(X) is the generating polynomial given by:

$$G(X) = X^{16} + X^{12} + X^{5} + 1.$$

The $X^{(n-16)} \cdot L(X)$ term has the effect of presetting the shift register to all "1" state prior to encoding.

A possible implementation of an encoder is described in Reference [5].

5.5.2 DECODING PROCEDURE

a. The error detection syndrome, S(X), is given by

$$S(X) = [(X^{16} \cdot C^{*}(X)) + (X^{n} \cdot L(X))] \text{ modulo } G(X)$$

where

— $C^{*}(X)$ is the received block, including the **FRAME ERROR CONTROL FIELD**, in polynomial form; and

— S(X) is the syndrome polynomial which will be zero if no error is detected and non-zero if an error is detected.

A possible implementation of a decoder is described in Reference [5].

INDEX